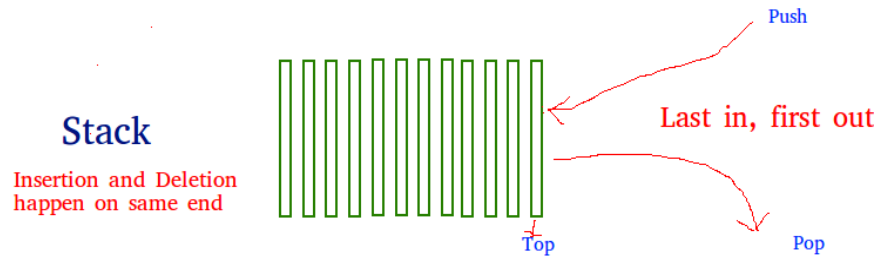
Date=06/08/2020
Lecture By=Shubham Joshi
Subject ⇒Stacks

IN PREVIOUS LECTURE **(QUICK RECAP) Date-05/08/2020**	**In Today's Lecture (Overview**
Problem Solving Session Question=1 Find the middle Element of linked List Question=2 Given A linked List You have to reverse it MCQs Questions for self practice	Stacks in python Question=1 Implement stack Using LINKED LIST Question=2Given an array Print Next Greater Element MCQs Questions For Self Practice

Stacks in python

A **stack** is a collection of objects that supports fast last-in, first-out (LIFO) semantics for inserts and deletes
Unlike lists or arrays, **stacks** typically don't allow for random access to the objects they contain.
The insert and delete operations are also often called push and pop.



The functions associated with stack are:

- **empty()** – Returns whether the stack is empty – Time Complexity : O(1)
- **size()** – Returns the size of the stack – Time Complexity : O(1)
- **top()** – Returns a reference to the top most element of the stack – Time Complexity : O(1)
- **push(g)** – Adds the element 'g' at the top of the stack – Time Complexity : O(1)
- **pop()** – Deletes the top most element of the stack – Time Complexity : O(1)

Practical implementation of Stacks in Python

```
# Python program to
# demonstrate stack implementation
# using list

stack = []

# append() function to push
# element in the stack
stack.append('a')
stack.append('b')
stack.append('c')

print('Initial stack')
print(stack)

# pop() fucntion to pop
# element from stack in
# LIFO order
print('\nElements poped from stack:')
print(stack.pop())
print(stack.pop())
print(stack.pop())

print('\nStack after elements are poped:')
print(stack)

# uncommenting print(stack.pop())
# will cause an IndexError
```

```
# as the stack is now empty
```

OutPut

```
Initial stack
['a', 'b', 'c']

Elements poped from stack:
c
b
a

Stack after elements are poped:
[]
```

Some words that are used in stacks

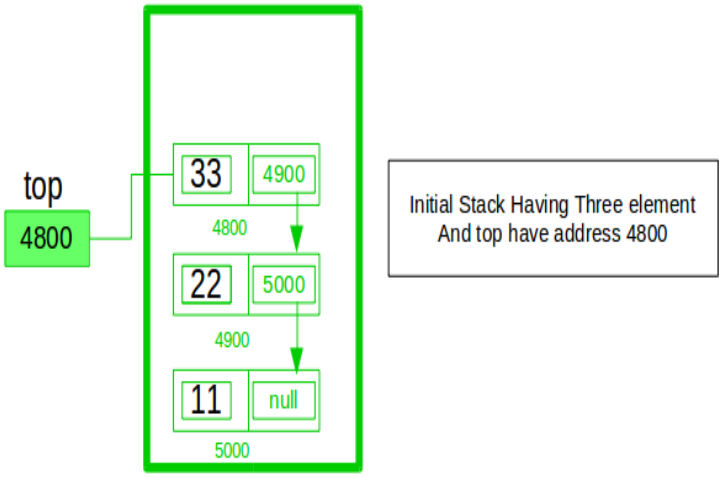
Add/push
=It will add Element to the stack

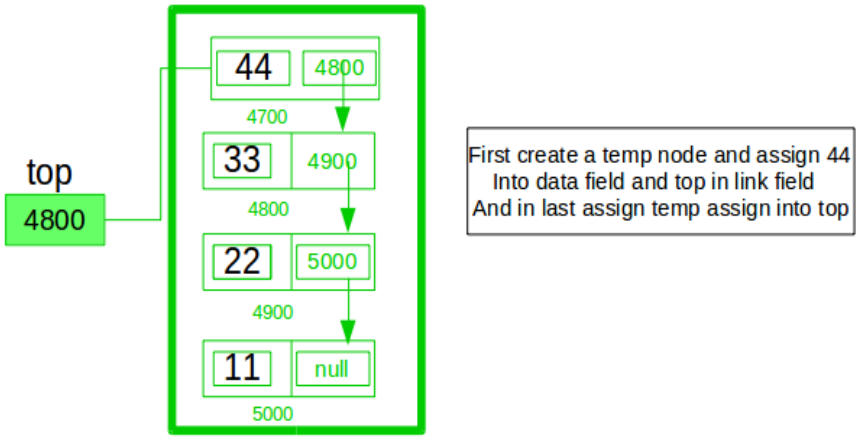
POP
=It removes last Inserted element from Stack in LIFO (last in first out) order

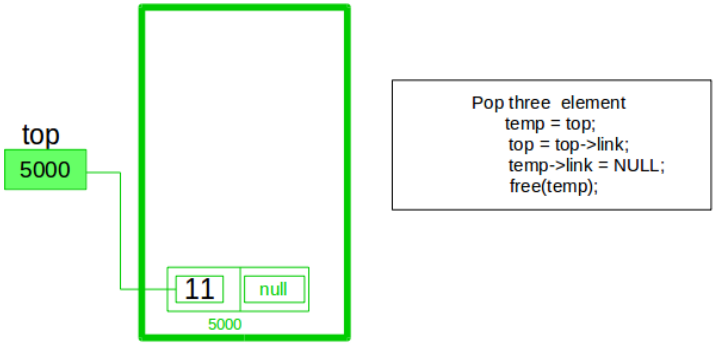
Peek
=it is used to know the top element of stacks

Problems that were solved in lecture

Question=1 Implement stack Using LINKED LIST







Code

```
class Node:

    # Class to create nodes of linked list
    # constructor initializes node automatically
```

```
    def __init__(self,data):
        self.data = data
        self.next = None

class Stack:

    # head is default NULL
    def __init__(self):
        self.head = None

    # Checks if stack is empty
    def isempty(self):
        if self.head == None:
            return True
        else:
            return False

    # Method to add data to the stack
    # adds to the start of the stack
    def push(self,data):

        if self.head == None:
            self.head=Node(data)

        else:
            newnode = Node(data)
            newnode.next = self.head
            self.head = newnode

    # Remove element that is the current head (start of the stack)
    def pop(self):

        if self.isempty():
            return None

        else:
            # Removes the head node and makes
```

```
            #the preceeding one the new head
            poppednode = self.head
            self.head = self.head.next
            poppednode.next = None
            return poppednode.data

    # Returns the head node data
    def peek(self):

        if self.isempty():
            return None

        else:
            return self.head.data

    # Prints out the stack
    def display(self):

        iternode = self.head
        if self.isempty():
            print("Stack Underflow")

        else:

            while(iternode != None):

                print(iternode.data,"->",end = " ")
                iternode = iternode.next
            return

# Driver code
MyStack = Stack()

MyStack.push(11)
MyStack.push(22)
MyStack.push(33)
MyStack.push(44)
```

```
# Display stack elements
MyStack.display()

# Print top element of stack
print("\nTop element is ",MyStack.peek())

# Delete top elements of stack
MyStack.pop()
MyStack.pop()

# Display stack elements
MyStack.display()

# Print top element of stack
print("\nTop element is ", MyStack.peek())
```

Output

```
E:\Study\Codes\if>C:/python/python.exe "e:/Study/Codes
44 -> 33 -> 22 -> 11 ->
Top element is  44
22 -> 11 ->
Top element is  22
```

Question=2**Given an array Print Next Greater Element**

Examples:

1. For any array, rightmost element always has next greater element as -1.
2. For an array which is sorted in decreasing order, all elements have next greater element as -1.

3. For the input array [4, 5, 2, 25}, the next greater elements for each element are as follows.

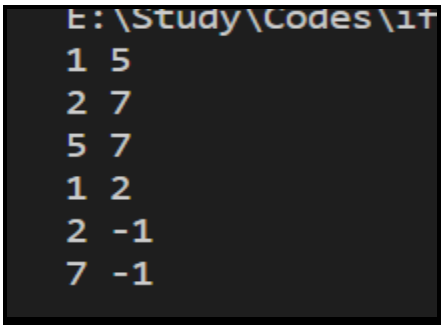
```
Element       NGE
   4      -->   5
   5      -->   25
   2      -->   25
   25     -->   -1
```

Code

```
class Stack:

    def __init__(self):

        self.__stack = []


    def push(self, x):

        self.__stack.append(x)


    def pop(self):

        return self.__stack.pop()


    def peek(self):

        return self.__stack[len(self.__stack) - 1]


    def is_empty(self):

        return len(self.__stack) == 0


def solve(array):
```

```
    s = Stack()
    idx = 0

    while idx < len(array):
        elem = array[idx]
        if s.is_empty():
            s.push(elem)
        else:
            while not s.is_empty() and s.peek() < elem :
                print(s.peek(), elem)
                s.pop()

            s.push(elem)
        idx += 1

    while not s.is_empty():
        print(s.peek(), -1)
        s.pop()

if __name__ == '__main__':
    solve([1, 5, 2, 7 , 1, 2])
```

Output



MCQs

1.what will be the output of following ? push(10) get_min() push(1) get_min() pop() get_min()

A.10 1 10

B.1 10 10

C.10 10 1

2.what is the best time complexity to find the next greater element using stack.2

A.O(nlogn)

B.O(n) + O(1) space

C.O(n) time and O(n) space

3.What will be the output of it ? push(10) push(2) push(3) peek() pop() push(15) pop() pop() pop()

A.3 3 15 2 10

B.3 15 2 10

C.3 15 3 2 10

D.3 2 15 10

4.What will be the output of the following ? push(5) pop(5) push(4) push(3) pop() pop()

A.3 5 4

B.5 4 3

C.5 3 4

Questions For Self Practice\CC FOR THE DAY

https://leetcode.com/problems/valid-parentheses/

https://leetcode.com/problems/next-greater-element-ii/